TrailsCommunity

Rapport de projet - Assitant Offroad

BELLANGER Stéphen MONNIER Ysée DESHORS Yann

Novembre 2016 - Decembre 2016

Table des matières

Table des figures

Première partie

Client TrailsCommunity

Chapitre 1

Introduction

Cette section fournit une vue d'ensemble de tout ce qui est inclus dans ce document SRS. De plus, l'objectif de ce document est décrit et une liste de définitions est fournie.

1.1 Objectifs

//A refaire : mal expliqué TrailsCommunity est une application Android développé en Java sous IDE Android Studio. Elle est composé d'un serveur développé en Ruby (Framework Ruby On Rails). La porté de l'application est l'ensemble des utilisateurs ayant pour objectifs de réalisé des activités en pleines air. Nous estimons une tranche d'âge entre 18 à 80 ans. Toutefois, elle peut aussi être utilisé par des professionelles ou des personnes voulant organisé ou suppervisé des sorties. La version sera disponible sur le Google play gratuitement.

<Identify the product whose software requirements are specified in this document, including the revision or release number. Describe the scope of the product that is covered by this SRS, particularly if this SRS describes only part of the system or a single subsystem.>

1.2 Conventions

Le logo de l'application souhaite représenter la nature est les activités en pleines airs. La typographie du nom TrailsCommunity est : ...

//Faire une charte graphique -> ajouter a la TODO liste

La liste des fonctionnalités et leurs priorités sont présentés plus tard dans ce rapport. Celle-ci ont été déterminé avec l'ensemble des avis des membres du groupe. Le classement est rélisé par une échelle de 3 graduation : bas, moyen, haut.

<Describe any standards or typographical conventions that were followed when writing this SRS, such as fonts or highlighting that have special significance. For example, state whether priorities for higher-level requirements are assumed to be inherited by detailed requirements, or whether every requirement statement is to have its own priority.>

1.3 Intended Audience and Reading Suggestions

Ce document est destiné dans un premier temps au correcteur désigné de l'Université de Toulon pour la correction de ce projet. Cependant, il sera alors disponible en public dans le dépot GitHub public utilisé durant le développement de l'application. L'ensemble de ce rapport est organisé de la manière suivant :

1. la présentation du client, sa la pré-conception et de ca conception UML.
2. puis vient la modélisation du serveur, ses conventions et l'ensemble de ses actions.
3. pour finir une annexe avec l'ensemble des diagrammes nécessaire au développement de l'ensemble de l'application qui n'on pu être présenté dans les partie du client et du serveur.

<Describe the different types of reader that the document is intended for, such as developers, project managers, marketing staff, users, testers, and documentation writers. Describe what the rest of this SRS contains and how it is organized. Suggest a sequence for reading the document, beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.>

1.4 Description du projet

L'objectif de ce projet est de réaliser une application Android d'assistant offroad. L'application créé devra permettre d'enregistrer, de visualiser et de partager des positions GPS entre un groupe d'utilisateur. Cette application pourra donc être utilisé dans le cadre d'activité en groupe et en plein air(randonnées, painball, vélo). De plus, chaque utilisateur pourra visualiser différentes statistiques en rapport avec ses activités réalisées précédemment. Il est plus que nécessaire pour ce genre d'application d'avoir un temps de réponse rapide du serveur. C'est pour cela que la stratégie principal du groupe est de réalisé un serveur dédié uniquemment pour l'application. De plus, les éxigences sur les règles de codage sont appliqué sur le client.

<Provide a short description of the software being specified and its purpose, including relevant benefits, objectives, and goals. Relate the software to corporate goals or business strategies. If a separate vision and scope document is available, refer to it rather than duplicating its contents here.>

1.5 Répartition du travail

La majorité de la phase de conception a été réalisée à trois : les spécifications et la modélisation de l'axe fonctionnel ont été faites par le groupe au complet. Les diagrammes de séquence et d'activité ont été répartis entre nous, puis nous les avons tous passé en revue et modifié jusqu'à que nous soyons tous d'accord sur l'ensemble des diagrammes et des idées qu'elles véhiculent.

Lors de la phase de développement, nous avons réparti les rôles de la manière suivante :

— Ysée : Conception UMl, développement du serveur, développement de l'application mobile

— Stéphen : Conception UML, développement de l'application mobile, rapport

— Yann : Conception UML, rapport

Toutefois, vu l'objectif du projet, nous avons tous participé au développement du client, mais à un niveau d'implication différent en fonction des compétences des membres du groupe.

1.6 Definitions

1.6.1 Termes

session une session est une activité disponible dans l'application

visiteur personne n'étant pas identifier

utilisateur personne identifié à l'application. Il possède les choix de créer ou de rejoindre une session

organisateur utilisateur ayant créé une session et pouvant la gérer

participant utilisateur qui a rejoint une session

waypoint désigne un point de la route a atteindre ou doit avoir lieu un changement de cap

1.7 Références

The Institute of Electrical and Electronic Engineer NY USA IEEE Recommented Practice for SRS, `http://www.utdallas.edu/~chung/RE/IEEE830-1993.pdf`

Chalmers IEEE standards - SRS Example, 2010, `http://www.cse.chalmers.se/~feldt/courses/reqeng/examples/srs_example_2010_group2.pdf`

Droid5 Informatics Pvt Ltd androidhive, 2016, `http://www.androidhive.info`

Oracle Java 8 Documentation, 2016, `https://docs.oracle.com/javase/8/docs/api/`

Google Android Documentation, 2016, `https://developer.android.com/guide`

James Britt Ruby Doc, 2016, `http://ruby-doc.org`

Rails Community Ruby On Rails Documentation, 2016, `http://rubyonrails.org`

<List any other documents or Web addresses to which this SRS refers. These may include user interface style guides, contracts, standards, system requirements specifications, use case documents, or a vision and scope document. Provide enough information so that the reader could access a copy of each reference, including title, author, version number, date, and source or location.>

Chapitre 2

Description générale

2.1 Perspective du produit

Est ce que c'est pertinent ? On en parle deja dans "Intended Audience and Reading Suggestions" et "Objectifs"

<Describe the context and origin of the product being specified in this SRS. For example, state whether this product is a follow-on member of a product family, a replacement for certain existing systems, or a new, self-contained product. If the SRS defines a component of a larger system, relate the requirements of the larger system to the functionality of this software and identify interfaces between the two. A simple diagram that shows the major components of the overall system, subsystem interconnections, and external interfaces can be helpful.>

2.2 Fonctions du produit

Meme chose ici. Voir -> "Project Scope Description du projet"

<Summarize the major functions the product must perform or must let the user perform. Details will be provided in Section 3, so only a high level summary (such as a bullet list) is needed here. Organize the functions to make them understandable to any reader of the SRS. A picture of the major groups of related requirements and how they relate, such as a top level data flow diagram or object class diagram, is often effective.>

2.3 Classes et caractéristiques des utilisateurs

<Identify the various user classes that you anticipate will use this product. User classes may be differentiated based on frequency of use, subset of product functions used, technical expertise, security or privilege levels, educational level, or experience. Describe the pertinent characteristics of each user class. Certain requirements may pertain only to certain user classes. Distinguish the most important user classes for this product from those who are less important to satisfy.>

2.4 Environnement d'exploitation

<Describe the environment in which the software will operate, including the hardware platform, operating system and versions, and any other software components or applications with which it must peacefully coexist.>

2.5 Contraintes de conception et d'implémentation

<Describe any items or issues that will limit the options available to the developers. These might include : corporate or regulatory policies ; hardware limitations (timing requirements, memory requirements) ; interfaces to other applications ; specific technologies, tools, and databases to be used ; parallel operations ; language requirements ; communications protocols ; security considerations ; design conventions or programming standards (for example, if the customer's organization will be responsible for maintaining the delivered software).>

2.6 Documentation utilisateur

<List the user documentation components (such as user manuals, on-line help, and tutorials) that will be delivered along with the software. Identify any known user documentation delivery formats or standards.>

2.7 Hypothèses et dépendances

<List any assumed factors (as opposed to known facts) that could affect the requirements stated in the SRS. These could include third-party or commercial components that you plan to use, issues around the development or operating environment, or constraints. The project could be affected if these assumptions are incorrect, are not shared, or change. Also identify any dependencies the project has on external factors, such as software components that you intend to reuse from another project, unless they are already documented elsewhere (for example, in the vision and scope document or the project plan).>

Chapitre 3

Exigences de l'interface externe

3.1 Interfaces des utilisateurs

<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>

3.2 Interfaces logicielles

<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>

3.3 Interfaces de communication

<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>

Chapitre 4

Caractéristiques du système

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

4.1 Fonction système 1

<Don't really say "System Feature 1." State the feature name in just a few words.>

4.1.1 Description et priorité

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

4.1.2 Séquence de stimulation / réponse

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

4.1.3 Exigences fonctionnelles

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use "TBD" as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1 : REQ-2 :

4.2 System Feature 2 (and so on)

Chapitre 5

Autres exigences non fonctionnelles

5.1 Exigences de performance

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

5.2 Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product's design or use. Define any safety certifications that must be satisfied.>

5.3 Exigences de sécurité

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

5.4 Attributs de qualité du logiciel

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are : adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

Chapitre 6

Autres exigences

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

6.1 Annexe A : Glossaire

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>

6.2 Annexe B : Modèles d'analyse

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>

6.3 Annexe C : Liste à déterminer

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>